

(51)Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)
B 2 5 J 13/00		B 2 5 J 13/00	A 3 F 0 5 9
9/06		9/06	D 3 F 0 6 0
B 6 5 G 47/90		B 6 5 G 47/90	B 3 F 0 7 2
G 0 6 T 1/00	3 0 0	G 0 6 T 1/00	3 0 0 5 B 0 5 7

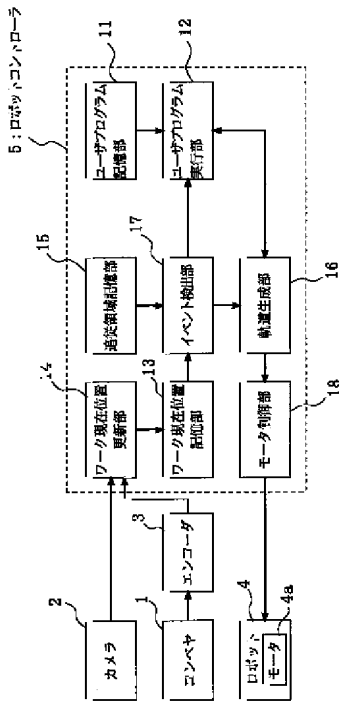
審査請求 未請求 請求項の数28 O L (全 15 頁)

(21)出願番号	特願2000-392392(P2000-392392)	(71)出願人	000002369 セイコーエプソン株式会社 東京都新宿区西新宿2丁目4番1号
(22)出願日	平成12年12月25日 (2000. 12. 25)	(71)出願人	000002325 セイコーインスツルメンツ株式会社 千葉県千葉市美浜区中瀬1丁目8番地
		(72)発明者	奥山 正幸 長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内
		(74)代理人	100061273 弁理士 佐々木 宗治 (外3名)

最終頁に続く

(54)【発明の名称】 ロボット制御方法及び該方法を適用したロボットコントローラ

(57)【要約】
【課題】 コンベヤの移動に伴うワークの現在位置の演算処理に際し、ワークの移動経路によらず演算量の低減を図ることを可能にすると共に、ワークに対する動作の記述及び直感的な教示を可能にするロボット制御方法及びロボットコントローラを提供する。
【解決手段】 ワークの現在位置をコンベヤ1の座標系で逐次更新するようにして、ワークを追従するためのロボット4の軌道は、ワークの位置をコンベヤ1の座標系からロボット4の座標系に変換することにより生成する。



【特許請求の範囲】

【請求項1】 運搬装置によって搬送される移動物体を追従して該移動物体に所定の動作を行うロボットの制御方法であって、

移動物体を検出し、その検出結果に基づいて移動物体の運搬装置の座標系における位置を取得し、その位置と運搬装置の動作量に基づいて移動物体の位置を運搬装置の座標系で逐次更新し、前記移動物体を追従するためのロボットの軌道を、前記移動物体の位置を運搬装置の座標系からロボットの座標系に変換することによって生成することを特徴とするロボット制御方法。

【請求項2】 前記運搬装置の座標系は、移動物体の移動方向に平行なX軸と、該X軸と共に運搬装置の搬送面を構成するY軸と、X軸及びY軸に直交するZ軸で構成されることを特徴とする請求項1記載のロボット制御方法。

【請求項3】 指令位置データが運搬装置の座標系で設定された動作命令が与えられると、前記動作命令の指令位置データで特定される移動物体の追従を開始することを特徴とする請求項1又は請求項2記載のロボット制御方法。

【請求項4】 運搬装置の座標系で予め追従領域を設定しておき、該追従領域を特定する追従領域座標と移動物体の現在位置とに基づいて移動物体が追従領域内にあるか否かを判断し、移動物体が追従領域内にあると判断した場合に、移動物体到達イベントを発行し、該移動物体到達イベントが発行されたときに、与えられた動作命令の指令位置データで特定される移動物体の追従を開始することを特徴とする請求項1乃至請求項3の何れかに記載のロボット制御方法。

【請求項5】 前記移動物体が追従領域内にあるか否かの判断を定期的に行うことを特徴とする請求項4記載のロボット制御方法。

【請求項6】 前記追従領域は、運搬装置の座標系のX軸に平行な2つの線と、X座標値が一定な2つの線とで特定される領域であることを特徴とする請求項4又は請求項5記載のロボット制御方法。

【請求項7】 移動物体を追従するための軌道は、追従開始時における移動物体の位置に向かう方向の軌道に追従方向の軌道を各関節の角度毎に加算することによって生成し、前記追従方向の軌道は、移動物体の現在位置と所定周期前の移動物体の位置とをそれぞれ運搬装置の座標系からロボットの座標系に変換し、これらを更に関節角度座標系に変換して取得した現在位置における関節角度と、所定周期前の関節角度との差分を取ることにより求めることを特徴とする請求項1乃至請求項6の何れかに記載のロボット制御方法。

【請求項8】 移動物体が追従領域内にあるか否かを判断した際に、追従領域内にないと判断した場合には追従停止イベントを発行して、該追従停止イベントに対応す

る処理を開始することを特徴とする請求項7記載のロボット制御方法。

【請求項9】 前記追従停止イベントに対応する処理は、追従方向の軌道の生成を停止する処理であることを特徴とする請求項8記載のロボット制御方法。

【請求項10】 指令位置データがロボットの座標系で設定された動作命令が与えられると、前記追従方向の軌道の生成を停止して、前記指令位置への軌道を生成することを特徴とする請求項7乃至請求項9の何れかに記載のロボット制御方法。

【請求項11】 複数台の運搬装置を配置する場合に、運搬装置毎にそれぞれ運搬装置の座標系を設定することを特徴とする請求項1乃至請求項10の何れかに記載のロボット制御方法。

【請求項12】 複数台のロボットを配置する場合に、ロボット毎にそれぞれ運搬装置の座標系及び追従領域を設定することを特徴とする請求項1乃至請求項10の何れかに記載のロボット制御方法。

【請求項13】 前記運搬装置は直動コンベヤであることを特徴とする請求項1乃至請求項12の何れかに記載のロボット制御方法。

【請求項14】 前記運搬装置は円弧コンベヤ又はターンテーブルであり、前記運搬装置の座標系は、コンベヤの回転方向に平行で、回転角度を軸方向の変位とするX軸と、回転中心を通るY軸と、前記X軸及びY軸に直交するZ軸で構成することを特徴とする請求項1乃至請求項12の何れかに記載のロボット制御方法。

【請求項15】 運搬装置によって搬送される移動物体に追従して該移動物体に所定の動作を行うロボットのコントローラであって移動物体の現在位置が格納される移動物体現在位置記憶部と、

移動物体を検出する検出部と、
前記検出部の検出結果に基づいて移動物体の運搬装置の座標系における位置を取得し、その位置と運搬装置の動作量とに基づいて移動物体の現在位置を算出し、算出した位置データで前記移動物体現在位置記憶部を更新する移動物体現在位置更新部と、
前記移動物体現在位置記憶部の移動物体の現在位置を運搬装置の座標系からロボットの座標系に変換することによって移動物体に追従するためのロボットの軌道を生成する軌道生成部とを備えたことを特徴とするロボットコントローラ。

【請求項16】 前記運搬装置の座標系は、移動物体の移動方向に平行なX軸と、該X軸と共に運搬装置の搬送面を構成するY軸と、X軸及びY軸に直交するZ軸で構成されることを特徴とする請求項15記載のロボットコントローラ。

【請求項17】 動作命令が記述されたユーザプログラムを実行し、動作命令の指令位置データが運搬装置の座標系で記述されているのか否かを判断するユーザプログ

ラム実行部を更に備え、前記軌道生成部は、前記ユーザプログラム実行部において前記指令位置データが運搬装置の座標系で記述されていると判断されたとき、該動作命令の指令位置データで特定される移動物体に追従するための軌道を生成することを特徴とする請求項15又は請求項16記載のロボットコントローラ。

【請求項18】 運搬装置の座標系で追従領域を設定するための追従領域座標が格納される追従領域記憶部と、移動物体が追従領域内にあるか否かを前記現在位置記憶部に格納された移動物体の現在位置と前記追従領域記憶部に格納された追従領域座標とに基づいて判断し、領域内にあると判断した場合には移動物体到達イベントを発行し、領域内にないと判断した場合には追従停止イベントを発行するイベント検出部とを更に備え、前記軌道生成部は、移動物体到達イベントが発行されたときに、動作命令の指令位置データで特定される移動物体に追従するための軌道を生成することを特徴とする請求項15乃至請求項17の何れかに記載のロボットコントローラ。

【請求項19】 前記イベント検出部は、移動物体が領域内にあるか否かの判断を定期的に行うことを特徴とする請求項18記載のロボットコントローラ。

【請求項20】 前記追従領域は、運搬装置の座標系のX軸に平行な2つの線と、X座標値が一定な2つの線とで特定される領域であることを特徴とする請求項18又は請求項19記載のロボットコントローラ。

【請求項21】 前記軌道生成部は、前記移動物体に追従する軌道を、追従開始時における移動物体の位置に向かう方向の軌道に、追従方向の軌道を加算することによって生成し、前記追従方向の軌道は、移動物体の現在位置と所定周期前の移動物体の位置とをそれぞれ運搬装置の座標系からロボット座標系に変換し、これらを更に関節角度座標系に変換して取得した現在位置における関節角度と、所定周期前の関節角度との差分を取ることで求めらることを特徴とする請求項15乃至請求項20の何れかに記載のロボットコントローラ。

【請求項22】 前記ユーザプログラム実行部は、イベント検出部から追従停止イベントが発行されると、該追従停止イベントに対応する処理の実行を開始することを特徴とする請求項21記載のロボットコントローラ。

【請求項23】 前記追従停止イベントに対応する処理は、追従方向の軌道の生成を停止する処理であることを特徴とする請求項22記載のロボットコントローラ。

【請求項24】 前記軌道生成部は、ユーザプログラム実行部において動作命令の指令位置データがロボットの座標系であると判断された場合、前記追従方向の軌道の生成を停止して、前記指令位置への軌道を生成することを特徴とする請求項21乃至請求項23の何れかに記載のロボットコントローラ。

【請求項25】 複数台の運搬装置を配置する場合に、運搬装置毎にそれぞれ運搬装置の座標系を設定する

ことを特徴とする請求項15乃至請求項24の何れかに記載のロボットコントローラ。

【請求項26】 複数台のロボットを配置する場合には、ロボット毎にそれぞれ運搬装置の座標系及び追従領域を設定することを特徴とする請求項15乃至請求項24の何れかに記載のロボットコントローラ。

【請求項27】 前記運搬装置は直動コンベヤであることを特徴とする請求項15乃至請求項26の何れかに記載のロボットコントローラ。

【請求項28】 前記運搬装置は円弧コンベヤ又はターンテーブルであり、前記運搬装置の座標系は、コンベヤの回転方向に平行で、回転角度を軸方向の変位とするX軸と、回転中心を通るY軸と、前記X軸及びY軸に直交するZ軸で構成することを特徴とする請求項27記載のロボットコントローラ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明はコンベヤ等によって搬送される移動物体を追跡し、該移動物体に対して所定の動作を行うロボットの制御方法及びその方法を適用したロボットコントローラに関する。

【0002】

【従来の技術】従来、この種の移動物体（以下、ワークという）を追従するコンベヤトラッキングでは、ワークがロボットの可動範囲に到達したことを近接スイッチ等の外部センサで検出し、その検出信号に基づいてロボット手先の追従を開始させ、ワークに到達したときにそのワークに対して作業を行うように構成している。このような従来のコンベヤトラッキングでは、コンベヤ上に搬送された自動車の車体の溶接を行うアプリケーションを主としており、このため、コンベヤが低速で移動し、同時に処理するワーク数が少ない（大抵は1）ことを想定してシステムが構成されている。

【0003】

【発明が解決しようとする課題】ところが、近年ではコンベヤトラッキングに望まれるアプリケーションも多種に渡っており、コンベヤ上をある程度の速度をもって流れる複数のワークに対して作業を行わせたいという要望が発生してきた。しかしながら、上述のようにワーク数が少ないことを想定して構成されたシステムでは、この種のアプリケーションの実現は困難であった。

【0004】例えば、ワークの現在位置を更新するに際し、ワーク数に比例した計算量が必要となることから、仮にロボットがコンベヤに対して平行ではなく傾いて設置されていた場合、コンベヤの動作量に対してその傾いた角度の三角関数演算を行なって現在位置を算出する必要があるため、ロボット座標系（ロボットの設置ベースを原点とするZ軸を垂直軸とする直交座標系（X、Y、Z）とする。）においてX座標値及びY座標値の両方について演算が必要となり、その演算量は多大なものとな

る。このため、同時に処理できるワーク数に制限があったり、多数のワークを処理するための高速なCPUが必要となる等の問題が生じていた。

【0005】ところで、このようなコンベア上のワークをロボットに追従させるための動作プログラムにおいては、ロボット手先の目標位置をロボット座標系で設定するようにしており、このためコンベヤを基準とした位置指定（例えばワーク中心から上流側に5mmの位置や、コンベヤの幅方向に10mm上部側の位置等の指定）が困難で、プログラム記述が複雑化していた。

【0006】ところで、最近では、カメラによってワークを検出するようにしたシステムが登場しているが、この種のシステムは、カメラによってワークの位置や向きなども検出できることから、複数のワークがコンベヤ上に散在しているような場合にも適用可能なシステムとして期待されている。しかしながらこの種のシステムにおいても、上述のワーク現在位置の更新にあたっての計算量の問題については依然解決されていない。

【0007】また、これらの従来のシステムにおいては、ワークがロボットの動作範囲に到達したか、また動作範囲から逸脱していないか等をチェックするための処理をユーザにプログラミングさせるようにしており、このため、ユーザプログラム中にワークの現在位置が動作範囲にあるか否かを監視する処理ループを記述する必要があった。次の図はそのユーザプログラムの具体的一例を示している。

【0008】図20はユーザプログラムの具体的一例を示す図、図21は図20のユーザプログラムに対応した処理フローである。なお、両図において対応する処理部分には同一のステップ番号を示している。このユーザプログラムにおいては、まずワークがロボットの可動範囲内にあるのか否かを処理ループ（DO・・・LOOP）によってチェックし（S101）、可動範囲内にあれば追従軌道を生成してロボットをワークに追従させ（S102）、この追従軌道の生成をワークの現在位置が可動範囲を超えるか否かを処理ループ（REPEAT・・・UNTIL）でチェックしながら（S103）、ワークのハンドリングが終了するまで繰り返す（S104）、途中、ロボットが可動範囲を超えた場合には追従動作を停止させ、エラー処理を行わせるようにしている（S105）。

【0009】このように、従来技術においては、ワークがロボットの動作範囲（ハンドリング領域）に到達したか否かをユーザプログラムに基づいてチェックするようにしているため、そのプログラム作成においてユーザの負担を強いるものとなっており、またプログラムが複雑化してプログラムの可読性を低下させていた。

【0010】また、ロボットが追従動作を行っている最中に何らかの原因でプログラムが停止してしまった場合には、ロボット動作範囲内の他の装置に衝突してしまっ

たり、ロボット動作範囲を超えて動作を行おうとしてエラーとなるなどの不具合が発生する可能性があった。

【0011】また、カメラやロボット等の複数の装置を制御するロボットコントローラにおいては、ワークの位置取得に係るプログラムやロボットの動作プログラム等、複数のプログラムを並列処理して高速に実行できるマルチタスク機能が備えられている。ところが、ワークの到達や逸脱等、所定の間隔でチェックすれば十分な処理であっても、処理ループで実行されるために過剰に処理され、これにより他のプログラムの実行速度を低下させてしまい、マルチタスク機能の利点がいかされずロボットコントローラ全体の性能低下を招いていた。

【0012】また、従来のハンドリングシステムは直動コンベヤを対象としたものが多く、ターンテーブルや円弧コンベヤ等、ワークの移動経路が曲線状になるコンベヤに対しての追従制御が困難であった。この課題を解消するものとして、特開昭60-221805号公報に記載されたものがあるが、この技術においても、ワークの現在位置の演算に係る演算量の問題については解決されていない。

【0013】本発明は、このような点に鑑みなされたもので、コンベヤの移動に伴うワークの現在位置の演算処理に際し、ワークの移動経路によらず演算量の低減を図ることを可能にすると共に、移動物体に対する動作の記述の容易化及び直感的な教示を可能にするロボット制御方法及び該方法を適用したロボットコントローラを提供することを第1の目的とする。

【0014】また、本発明は上記第1の目的に加え、ロボットのハンドリング領域（追従領域）を設定し、その追従領域とワークとの位置関係をチェックする機能を提供することで容易なプログラム記述及び処理速度の向上を図ることが可能なロボット制御方法及び該方法を適用したロボットコントローラを提供することを第2の目的とする。

【0015】

【課題を解決するための手段】（1）本発明の一つの態様に係るロボット制御方法は、運搬装置によって搬送される移動物体を追従して移動物体に所定の動作を行うロボットの制御方法であって、移動物体を検出し、その検出結果に基づいて移動物体の運搬装置の座標系における位置を取得し、その位置と運搬装置の動作量に基づいて移動物体の位置を運搬装置の座標系で逐次更新し、移動物体を追従するためのロボットの軌道を、移動物体の位置を運搬装置の座標系からロボットの座標系に変換することによって生成するものである。

【0016】（2）本発明の他の態様に係るロボット制御方法は、上記（1）において運搬装置の座標系を、移動物体の移動方向に平行なX軸と、X軸と共に運搬装置の搬送面を構成するY軸と、X軸及びY軸に直交するZ軸で構成したものである。

【0017】(3)本発明の他の態様に係るロボット制御方法は、上記(1)又は(2)において指令位置データが運搬装置の座標系で設定された動作命令が与えられると、動作命令の指令位置データで特定される移動物体の追従を開始するものである。

【0018】(4)本発明の他の態様に係るロボット制御方法は、上記(1)～(3)の何れかにおいて、運搬装置の座標系で予め追従領域を設定しておき、追従領域を特定する追従領域座標と移動物体の現在位置とに基づいて移動物体が追従領域内にあるか否かを判断し、移動物体が追従領域内にあると判断した場合に、移動物体到達イベントを発行し、移動物体到達イベントが発行されたときに、与えられた動作命令の指令位置データで特定される移動物体の追従を開始するものである。

【0019】(5)本発明の他の態様に係るロボット制御方法は、上記(4)において、移動物体が追従領域内にあるか否かの判断を定期的に行うものである。

【0020】(6)本発明の他の態様に係るロボット制御方法は、上記(4)又は(5)において、追従領域を、運搬装置の座標系のX軸に平行な2つの線と、X座標値が一定な2つの線とで特定される領域で構成したものである。

【0021】(7)本発明の他の態様に係るロボット制御方法は、上記(1)～(6)の何れかにおいて、移動物体を追従するための軌道を、追従開始時における移動物体の位置に向かう方向の軌道に追従方向の軌道を各関節の角度毎に加算することによって生成し、追従方向の軌道は、移動物体の現在位置と所定周期前の移動物体の位置とをそれぞれ運搬装置の座標系からロボットの座標系に変換し、これらを更に関節角度座標系に変換して取得した現在位置における関節角度と、所定周期前の関節角度との差分を取ることで求めたものである。

【0022】(8)本発明の他の態様に係るロボット制御方法は、上記(7)において、移動物体が追従領域内にあるか否かを判断した際に、追従領域内にないと判断した場合には追従停止イベントを発行して、追従停止イベントに対応する処理を開始するものである。

【0023】(9)本発明の他の態様に係るロボット制御方法は、上記(8)において、追従停止イベントに対応する処理を、追従方向の軌道の生成を停止する処理としたものである。

【0024】(10)本発明の他の態様に係るロボット制御方法は、上記(7)～(9)の何れかにおいて、指令位置データがロボットの座標系で設定された動作命令が与えられると、追従方向の軌道の生成を停止して、指令位置への軌道を生成するものである。

【0025】(11)本発明の他の態様に係るロボット制御方法は、上記(1)～(10)の何れかにおいて、複数台の運搬装置を配置する場合には、運搬装置毎にそれぞれ運搬装置の座標系を設定するものである。

【0026】(12)本発明の他の態様に係るロボット制御方法は、上記(1)～(10)の何れかにおいて、複数台のロボットを配置する場合には、ロボット毎にそれぞれ運搬装置の座標系及び追従領域を設定するものである。

【0027】(13)本発明の他の態様に係るロボット制御方法は、上記(1)～(12)の何れかにおいて運搬装置を直動コンベヤとしたものである。

【0028】(14)本発明の他の態様に係るロボット制御方法は、上記(1)～(12)の何れかにおいて運搬装置を円弧コンベヤ又はターンテーブルとし、運搬装置の座標系を、コンベヤの回転方向に平行で、回転角度を軸方向の変位とするX軸と、回転中心を通るY軸と、X軸及びY軸に直交するZ軸で構成したものである。

【0029】(15)本発明の一つの態様に係るロボットコントローラは、運搬装置によって搬送される移動物体に追従して移動物体に所定の動作を行うロボットのコントローラであって移動物体の現在位置が格納される移動物体現在位置記憶部と、移動物体を検出する検出部と、検出部の検出結果に基づいて移動物体の運搬装置の座標系における位置を取得し、その位置と運搬装置の動作量とに基づいて移動物体の現在位置を算出し、算出した位置データで移動物体現在位置記憶部を更新する移動物体現在位置更新部と、移動物体現在位置記憶部の移動物体の現在位置を運搬装置の座標系からロボットの座標系に変換することによって移動物体に追従するためのロボットの軌道を生成する軌道生成部とを備えたものである。

【0030】(16)本発明の他の態様に係るロボットコントローラは、上記(15)において、運搬装置の座標系を、移動物体の移動方向に平行なX軸と、X軸と共に運搬装置の搬送面を構成するY軸と、X軸及びY軸に直交するZ軸で構成したものである。

【0031】(17)本発明の他の態様に係るロボットコントローラは、上記(15)又は(16)において、動作命令が記述されたユーザプログラムを実行し、動作命令の指令位置データが運搬装置の座標系で記述されているのか否かを判断するユーザプログラム実行部を更に備え、軌道生成部は、ユーザプログラム実行部において指令位置データが運搬装置の座標系で記述されていると判断されたとき、動作命令の指令位置データで特定される移動物体に追従するための軌道を生成するものである。

【0032】(18)本発明の他の態様に係るロボットコントローラは、上記(15)～(17)の何れかにおいて、運搬装置の座標系で追従領域を設定するための追従領域座標が格納される追従領域記憶部と、移動物体が追従領域内にあるか否かを現在位置記憶部に格納された移動物体の現在位置と追従領域記憶部に格納された追従領域座標とに基づいて判断し、領域内にあると判断した場合には移動物体到達イベントを発行し、領域内にないと判断した場合には追従停止イベントを発行するイベント

検出部とを更に備え、軌道生成部は、移動物体到達イベントが発行されたときに、動作命令の指令位置データで特定される移動物体に追従するための軌道を生成するものである。

【0033】(19)本発明の他の態様に係るロボットコントローラは、上記(18)において、イベント検出部が、移動物体が領域内にあるか否かの判断を定期的に行うものである。

【0034】(20)本発明の他の態様に係るロボットコントローラは、上記(18)又は(19)において、追従領域を、運搬装置の座標系のX軸に平行な2つの線と、X座標値が一定な2つの線とで特定される領域としたものである。

【0035】(21)本発明の他の態様に係るロボットコントローラは、上記(15)～(20)の何れかにおいて、軌道生成部は、移動物体に追従する軌道を、追従開始時における移動物体の位置に向かう方向の軌道に、追従方向の軌道を加算することによって生成し、追従方向の軌道は、移動物体の現在位置と所定周期前の移動物体の位置とをそれぞれ運搬装置の座標系からロボット座標系に変換し、これらを更に関節角度座標系に変換して取得した現在位置における関節角度と、所定周期前の関節角度との差分を取ることににより求めるものである。

【0036】(22)本発明の他の態様に係るロボットコントローラは、上記(21)において、ユーザプログラム実行部は、イベント検出部から追従停止イベントが発行されると、追従停止イベントに対応する処理の実行を開始するものである。

【0037】(23)本発明の他の態様に係るロボットコントローラは、上記(22)において、追従停止イベントに対応する処理を、追従方向の軌道の生成を停止する処理としたものである。

【0038】(24)本発明の他の態様に係るロボットコントローラは、上記(21)～(23)の何れかにおいて、軌道生成部を、ユーザプログラム実行部において動作命令の指令位置データがロボットの座標系であると判断された場合に、追従方向の軌道の生成を停止して、指令位置への軌道を生成するようにしたものである。

【0039】(25)本発明の他の態様に係るロボットコントローラは、上記(15)～(24)の何れかにおいて、複数台の運搬装置を配置する場合には、運搬装置毎にそれぞれ運搬装置の座標系を設定するようにしたものである。

【0040】(26)本発明の他の態様に係るロボットコントローラは、上記(15)～(24)の何れかにおいて、複数台のロボットを配置する場合には、ロボット毎にそれぞれ運搬装置の座標系及び追従領域を設定するようにしたものである。

【0041】(27)本発明の他の態様に係るロボットコントローラは、上記(15)～(26)の何れかにおい

て、運搬装置を直動コンベヤとしたものである。

【0042】(28)本発明の他の態様に係るロボットコントローラは、上記(27)において、運搬装置を円弧コンベヤ又はターンテーブルとし、運搬装置の座標系を、コンベヤの回転方向に平行で、回転角度を軸方向の変位とするX軸と、回転中心を通るY軸と、X軸及びY軸に直交するZ軸で構成したものである。

【0043】上記(1)及び(15)によれば、運搬装置の座標系で移動物体の位置を更新することにより、ロボットの座標系で更新する場合に比べてその更新に係る演算が容易となり、また、移動物体に対する所定の動作を記述する際に、運搬装置の座標系を基準とした簡単かつ直感的な位置指定が可能でプログラム記述が容易となる。

【0044】上記(2)及び(16)によれば、運搬装置の座標系のX軸が移動物体の移動方向に平行に構成されているので、移動物体の位置はX座標値のみが変化する。このため、移動物体の位置はX座標値のみ更新すればよく、ロボット座標で更新する場合に比べて演算量を低減することが可能となる。

【0045】上記(3)及び(17)によれば、追従開始の指示を行わなくても運搬装置の座標系における指令位置データの動作命令を与えるだけでロボットに追従動作を開始させることが可能となるので、ユーザは運搬装置の動作を意識することなくプログラム記述が可能となる。

【0046】上記(4)、(5)、(6)、(18)、(19)、(20)によれば、運搬装置の座標系で予め追従領域を設定しておき、移動物体が追従領域内にあるか否かを追従領域座標と移動物体の現在位置とに基づき判断して、追従領域内にある場合には、移動物体到達イベントを発行し、移動物体到達イベントが発行されたとき、与えられた動作命令の指令位置データで特定される移動物体を追従する処理を開始するようにしたので、従来のようにユーザプログラム内で移動物体の位置を監視して処理ループを記述する必要がなくなり、プログラムの可読性の向上が図れると共に、処理速度を向上させることが可能となる。

【0047】上記(7)及び(21)によれば、移動物体の現在位置と所定周期前の移動物体の位置に基づき追従方向の軌道を生成しているので、運搬装置の速度ムラを補償した軌道を生成することが可能となる。

【0048】上記(8)及び(22)によれば、追従領域内にないと判断した場合に、追従停止イベントを発行して、その追従停止イベントに対応する処理を開始するので、例えばロボットが追従動作を行ったまま自分の動作範囲外にまで動作しようとしてエラーとなる等の不具合を回避するための動作プログラムを簡単に作成することが可能となる。即ち、ユーザは追従停止イベントに対応する処理として上記不具合回避動作を記述すれば良

い。

【0049】上記(9)及び(23)によれば、追従停止イベントが発行されたとき、追従方向の軌道の生成が停止されるので、ロボットが追従領域を逸脱して追従を続けることによる不具合を自動的に且つ確実に防止することが可能となる。

【0050】上記(10)及び(24)によれば、指令位置データがロボットの座標系で設定された動作命令が与えられたとき、追従方向の軌道の生成を停止して、指令位置への軌道が生成されるので、ユーザは目標位置(指令位置)が運搬装置外であれば、ロボットの座標系で指令位置データを記述することにより、自動的に追従動作を停止させてロボットを指令位置へ移動させる動作を実現することが可能となり、ロボットの現在の状態(追従中であるか否か)や運搬装置の動作を意識することなくプログラムを作成することが可能となる。

【0051】上記(11)及び(25)によれば、複数台の運搬装置を配置する場合には、運搬装置毎にそれぞれ運搬装置の座標系を設定するようにしたので、対応する座標系で指令位置データを設定した動作命令を記述することにより、一方の運搬装置から他方の運搬装置へ移動物体を移載するような作業も簡単にプログラムすることが可能となる。

【0052】上記(12)及び(26)によれば、複数台のロボットを配置する場合には、各ロボット毎にそれぞれ運搬装置の座標系及び追従領域を設定することにより、1台の運搬装置に複数台のロボットを配置することが可能となる。

【0053】上記(13)、(14)、(27)及び(28)によれば、運搬装置が直動コンベヤ、円弧コンベヤ、ターンテーブルの何れであっても、以上に説明した効果を奏することが可能となる。

【0054】

【発明の実施の形態】実施の形態1. 図1は本発明の実施の形態1のロボットコントローラを適用したハンドリングシステムの機能ブロック図、図2は本発明の実施の形態1のソフトウェア構成を示す図、図3は本発明の実施の形態1のハンドリングシステムの概略構成図である。図において、1は運搬装置としてのコンベヤ、2はコンベヤ1上を流れるワークを検出する検出部であるカメラ、3はコンベヤ1に取り付けられ、コンベヤ1の動作量を検出するためのエンコーダ、4はロボットで、これらコンベヤ1、カメラ2、エンコーダ3及びロボット4がロボットコントローラ5に接続されてハンドリングシステムが構成されている。

【0055】ロボットコントローラ5は、ユーザプログラムを記憶するユーザプログラム記憶部11、該ユーザプログラム記憶部11に記憶されたユーザプログラムを実行するユーザプログラム実行部12、カメラ2により認識されたワークの現在位置が格納されるワーク現在位

置記憶部13、該ワーク現在位置記憶部13のワークの現在位置をコンベヤ1の動作量に基づいて更新するワーク現在位置更新部14、追従領域が設定される追従領域記憶部15、作業ポイント(ロボット手先)の軌道(各関節の角度)を生成し、これをモータ指令値(パルス)に変換して後述のモータ制御部18に出力する軌道生成部16、追従領域に対するワークの事象(イベント)を検出するイベント検出部17、軌道生成部16からのモータ指令値に従って各関節のモータ4aを制御するモータ制御部18を備えた構成となっている。軌道生成部16及びイベント検出部17は外部タイマなどのハードウェアや後述のオペレーティングシステム(OS)により所定のサンプリングタイム毎に起動される。

【0056】また、ロボットコントローラ5のソフトウェア構成は、図2に示すようにユーザプログラムと、ロボットコントローラ5を制御し、イベント機能やマルチタスク機能等を有するオペレーティングシステム(OS)と、ロボット4やコンベヤ1の位置検出等を行う制御プログラムとを備えた構成となっている。

【0057】ここで、図1の各構成部の詳細な説明に先だって、本ロボットコントローラ5において導入するコンベヤ座標系(運搬装置の座標系)について説明する。図4はコンベヤ座標系の説明図である。コンベヤ座標系は、ワークの移動方向に平行なX軸と、このX軸と共に作業平面(コンベヤ1の搬送面)を構成する方向のY軸(直動コンベヤの場合、X軸に直交)と、X軸及びY軸に直交する方向のZ軸で構成される座標系として定義され、ロボット座標系と一意に変換可能となっている。なお、このコンベヤ座標系の原点は、ある時点でのワークの位置でもよいし、任意の位置に設定してもよい。また、ロボット座標系はロボット4の設置ベースを原点とする直交座標系である。

【0058】コンベヤ座標系における位置とロボット座標系における位置をユーザプログラム中に記述する場合には、以下のように記述する。

ロボット座標系の位置データ： $P_R = x, y, z, u, v, w$

コンベヤ座標系の位置データ： $P_C = xc, yc, zc, uc, vc, wc / CNV$

【0059】ここで、 x, y, z 及び xc, yc, zc はそれぞれ各座標軸成分の位置を表し、 u, v, w 及び uc, vc, wc はその位置での姿勢を表す成分である。また、座標値列の最後に「/CNV」と修飾することによりコンベヤ座標であることを示すようにしている。

【0060】なお、上述したようにコンベヤ座標系はロボット座標系と一意に変換可能であり、互いの変換式は変換行列Tを用いて以下のようにになっている。

【0061】

$$(x, y, z, u, v, w, 1) = T \cdot (xc, yc, zc, uc, vc, wc, 1)$$

$$(xc, yc, zc, uc, vc, wc, 1) = T^{-1} \cdot (x, y, z, u, v, w, 1)$$

【0062】このようなコンベヤ座標系の導入により、

コンベヤ動作に伴うワーク位置はX座標値のみが変化する。このようにワークの現在位置をコンベヤ座標系で演算することによりY座標値及びZ座標値の演算が省略でき、コンベヤ上を非常に多くのワークが流れる場合にも、ロボットコントローラに多大な負荷をかけることなく処理が可能となる。

【0063】図5は各座標系の階層構造を示す図である。ロボット座標系、関節角度座標系、モータパルス座標系は従来から利用されている座標系であり、ここで新たに導入するコンベヤ座標系を加えた各座標系は図5に示すような階層構造を成している。これらの各座標系は互いに座標変換可能となっており、ロボット4に対する

コンベヤ座標系で指令	: G O	xc, yc, zc, uc, vc, wc/CNV
ロボット座標系で指令	: G O	x, y, z, u, v, w
関節角度座標系で指令	: G O	J A (J1, J2, J3, J4, J5, J6)
モータパルス座標系で指令	: G O	P U L S E (p1, p2, p3, p4, p5, p6)

【0065】なお、関節角度座標系のJ1～J6は関節角度、モータパルス座標系のp1～p6はパルス数で、ロボットの各関節毎に指定される。なお、コンベヤ座標系においては位置データP_cで指定する以外に、例えばコンベヤ1上のワークの現在位置を示す変数「workpos」を用いて「G O workpos」と記述することも可能である。

【0066】ここで、図1の説明に戻る。カメラ2は、コンベヤ1上のワークを撮影しており、ワーク現在位置更新部14はその撮影画像からコンベヤ座標系におけるワークの位置を求めてその位置データをワーク現在位置記憶部13に格納する。この動作をワークの登録と位置づける。一方、エンコーダ3からはコンベヤパルスが出力されており、ワーク現在位置更新部14はそのコンベヤパルスに基づいてコンベヤ動作によるワークの移動量を算出し、ワーク現在位置記憶部13の位置データを逐次更新している。このワークの移動量は、撮影時のコンベヤパルス数と現在のコンベヤパルス数との差を求め、このパルス数の差と、予め設定された1コンベヤパルスあたりのコンベヤ座標系上での変化量との積を取ることにより求められる。ワーク現在位置更新部14は、このようにして算出したワークの移動量を、撮影時に算出された座標値に加算することによってコンベヤ座標におけるワーク現在位置を求め、ワーク現在位置記憶部13を更新する。ここで、Y座標及びZ座標については変化が無いので、ワーク現在位置更新部14はY座標、Z座標についての演算は省略し、X座標のみの演算を行う。

【0067】図6は図1の追従領域記憶部に設定される追従領域の説明図である。なお、図に示す座標系はコンベヤ座標系である（Z軸は図示せず）。この追従領域とはロボット4が追従動作を行う領域であり、ロボット4の動作範囲等を考慮してユーザによりコンベヤ座標系の座標値（追従領域座標）を以て設定される。すなわち、図6のハッチングで示す領域を指定する場合、a（x a, y a, 0）、b（x b, y b, 0）の2点の追従領

指令値は、何れの座標系で指令された場合においても、順次その下層に位置する座標系に変換され、最終的にはモータパルス座標系を基準としたパルスに変換されてモータ制御部18に出力されるようになっている。

【0064】次に示す命令文は各座標系での動作命令G Oの記述形式の一例を示している。ユーザは、ロボット手先の目標位置がコンベヤ1上の場合（例えば移動中のワークを指定する場合）にはコンベヤ座標系における位置データP_cを用いて動作命令を記述し、コンベヤ1外（例えばロボット脇に設置されたテーブル上）の場合には、ロボット座標系における位置データP_rを用いて動作命令を記述する。

域座標を追従領域設定値として追従領域記憶部15に設定する。

【0068】図7は図1のイベント検出部の動作フローである。イベント検出部17は、サンプリング間隔で呼び出されると、まず、ワーク現在位置記憶部13を参照してワークが登録されているか否かをチェックし（S1）、登録されている場合、ワーク現在位置と追従領域記憶部15の追従領域設定値とに基づいてワークが追従領域内にあるか否かをチェックする（S2）。ワークが追従領域内にある場合には、ワーク到達イベントを発行し（S3）、ワークが追従領域内になく、現在ロボット4が追従中であれば（S4）、追従停止イベントを発行して（S5）、処理を終了する。

【0069】図8は図1のユーザプログラム実行部の動作フローを示す図である。ユーザプログラム実行部12は、ユーザプログラムに記述された命令文を1行読み込み（S11）、読み込んだ命令が動作命令であるか否かを判断し（S12）、動作命令でないと判断した場合には、その命令に基づく処理を行い（S13）、動作命令であると判断した場合には、動作命令の指令位置データがコンベヤ座標系かロボット座標系かをチェックして（S14）、コンベヤ座標系であればフラグ(flag)を1に設定し（S15）、ロボット座標系であればフラグ(flag)を0に設定して（S16）、動作命令の登録を行う（S17）。この動作命令の登録とは、軌道生成部16での軌道生成処理に必要なデータを算出し、その算出データを軌道生成部16に登録する動作を行うものであり、後で詳述する。

【0070】ユーザプログラム実行部12は動作命令の登録後、軌道生成部16からの動作命令終了通知の受信待ち状態となる（S18）。即ち、動作命令で指令された位置へロボット手先が到達するのを待ち、その到達が軌道生成部16からの動作命令終了通知によって通知されると、ここで処理した命令がプログラムの最後かどうか

かを判断し (S19)、最後でなければステップ S11 に戻ってプログラムが終了するまで以上の処理を繰り返す行う。

【0071】ここで、動作命令の登録処理について説明する。この処理は、ロボット手先を現在位置から動作命令における指令位置（すなわち追従動作開始時のワーク位置）に移動させるのに要する関節角度移動量 ΔJ と、ロボット手先を現在位置から指令位置に移動させるのに要する時間 T とを算出して軌道生成部 16 に登録する処理を行うものである。

【0072】ここで、関節角度移動量 ΔJ は、ロボット 4 の現在の関節角度と、ロボット手先が指令位置に位置したときの関節角度との差分をとることにより求められる。図 9 はワークの位置と、ロボット 4 の関節角度との関係説明図である。図 9 においてロボットの現在位置を P_0 、指令位置をコンベヤ 1 上の位置 P_1 とすると（なお、 P_0 はロボット座標、 P_1 はコンベヤ座標とする）、この場合の関節角度移動量 ΔJ は、ワーク現在位置 P_1 をロボット座標系に変換し、更に関節座標系に変換して求めた関節角度座標 J_1 と、同様にロボット 4 の

$$J_s(t) = J_0 + \Delta J \cdot (t - t_0) / T \quad (t_0 \leq t \leq t_0 + T)$$

【0077】ここで、 J_0 はロボット 4 の現在位置 P_0 での関節角度、 t_0 は追従開始時の時刻である。

【0078】次いで、軌道生成終了か否かを時刻 t に基づいて判断し (S23)、現時刻 t が $t_0 + T$ 以内であれば軌道生成終了ではないと判断して、続いて指令位置がコンベヤ座標、ロボット座標の何れで与えられているかをフラグに基づいてチェックし (S25)、フラグが 1 の場合にはコンベヤ座標と判断して追従方向の軌道、即ちコンベヤ動作を考慮した追従動作分の関節角度移動量 $\Delta J_t(t)$ を算出する (S26)。

【0079】ここで、追従動作分の関節角度移動量 $\Delta J_t(t)$ の算出方法について上述の図 9 を参照しながら説明する。図 9 において、時刻 t でのコンベヤ座標におけるワーク位置を $P_t(t)$ 、このワークの 1 サンプル前の位置を $P_t(t-1)$ とする。追従動作分の関節角度移動量 $\Delta J_t(t)$ は、1 サンプルにおける関節角度の差分であり、位置 $P_t(t)$ 及び $P_t(t-1)$ をそれぞれコンベヤ座標系からロボット座標系に変換し、更に関節角度座標系に変換して取得した関節角度 $J_t(t)$ 及び $J_t(t-1)$ に基づいて次式で算出される。

$$\Delta J_t(t) = J_t(t-1) - J_t(t)$$

【0081】この関節角度移動量 $\Delta J_t(t)$ は次のサンプルにおける追従方向のロボット 4 の軌道として利用され、これによりコンベヤ 1 の動作量（コンベヤ 1 の速度）を考慮した目標軌道が生成されることになる。

【0082】軌道生成部 16 は、以上に算出した指令位置方向の関節角度 $J_s(t)$ と追従方向の関節角度移動量 $\Delta J_t(t)$ を各関節の成分毎に加算することによりサンプリングタイム毎の目標軌道である目標関節角度 J

現在位置 P_0 を関節角度座標系に変換して求めた関節角度座標 J_0 とを用いて次式で算出される。

$$\Delta J = J_1 - J_0$$

【0074】また、時間 T であるが、これは各関節のモータ制約等により予め設定された計算方法によって算出する。そして、このようにして算出した関節角度移動量 ΔJ 及び時間 T を動作命令として軌道生成部 16 に登録する。

【0075】図 10 は図 1 の軌道生成部の動作フローを示す図である。軌道生成部 16 は、サンプリング間隔で呼び出されると、まず動作命令が登録されているか否かをチェックし (S21)、登録されている場合、即ちロボット 4 の動作が指示されている場合には、その動作を実現するための軌道を生成する。その軌道生成として、まずロボット手先を現在位置から動作命令の指令位置に移動させるための指令位置方向（追従動作開始時におけるワーク位置に向かう方向）の軌道である関節角度 $J_s(t)$ を計算する (S22)。この関節角度 $J_s(t)$ は上述の ΔJ 、 T を用いた次式により算出される。

$$\Delta J = J_1 - J_0$$

【0076】 $J_s(t) = J_0 + \Delta J \cdot (t - t_0) / T$ (S27)。すなわち、次式で求められる。

$$J(t) = J_s(t) + \Delta J_t(t)$$

【0084】そして、軌道生成部 16 は関節角度 $J(t)$ をモータパルス座標に変換してモータ制御部 18 に出力する (S27)。一方、ステップ S25 においてロボット座標系であると判断した場合には、ステップ S26 の処理は行わずにステップ S27 の処理に移行する。すなわち、この関節角度移動量 $\Delta J_t(t)$ の演算を省略することによって追従動作が停止され、指令位置に向かう軌道のみが生成されることになる。このように、軌道生成部 16 は、指令位置の座標系に応じた軌道を逐次生成し、その目標関節角度をパルス座標に変換して出力する処理をサンプリングタイム周期で繰り返すように構成されている。

【0085】なお、ステップ S23 において時刻 $t_0 + T$ を経過し、ワークのハンドリングが終了している場合には、軌道生成終了と判断して動作命令終了をユーザプログラム実行部に通知する (S24)。時刻 $t_0 + T$ が経過しても追従停止が指示されない限り、追従方向の軌道 $\Delta J_t(t)$ の生成は続けられ、ロボット 4 は追従動作を継続する。ここで、例えば追従動作を行ったままワークが追従領域から逸脱したような場合には、イベント検出部 17 から追従停止イベントが発行されるので、この追従停止イベントの発行を待って追従動作を停止させるプログラムを作成することにより、追従方向の軌道の生成が停止され、追従領域を超えて追従動作を行ってしまうといったエラーを防止することが可能となる。

【0086】このように構成することにより、ユーザ

は、ロボット手先にコンベヤ1上のワークを追従させる場合には、ワークの位置をコンベヤ座標系で指定することにより（例えばワークの現在位置を示す関数workposで指定）、ロボット4に自動的に追従動作を開始させることが可能となり、また、コンベヤ1上のワークをハンドリング後、そのワークを例えばロボット脇のテーブル上に整列させたい場合には、テーブル上の目標位置をロボット座標系で指定することにより、自動的に追従動作を停止して、指定された目標位置へロボット手先を移動させる動作を行わせることが可能となる。従って、ユーザはコンベヤ1の動作を意識することなく、止まっているワークに対する処理を作成するように容易にプログラムを作成することが可能となる。次の図は、そのユーザプログラムの一例を示している。

【0087】図11は本発明の実施の形態のユーザプログラムの一例を示す図、図12は図11のタスク1及びタスク2の処理フローチャートで、両図において対応する処理部分には同一のステップ番号を示している。このプログラムは、メインタスク(FUNCION MAIN)において、他のファンクションの実行を指示する命令(XQT)により1行目においてタスク1を、2行目においてタスク2を起動させている。タスク1においては、ワークの到達を待ち(S31)、イベント検出部17からワーク到達イベントが発行され、ワークが到達したことを認識すると、ロボットにそのワークを追従させ(S32)、ハンドリングが終了すると(S33)、位置P2(ロボット座標系における位置)にロボット手先を移動させる(S34)処理を行うものである。また、タスク2においては、1行目において追従停止イベントを待ち(S41)、イベント検出部17から追従停止イベントが発行され、すなわちワークが追従領域から逸脱した場合、2行目においてタスク1を停止させ(S42)、3行目において追従動作を停止させ(S43)、4行目においてエラー処理を行う(S44)ようにしたものである。

【0088】次に本実施の形態1の動作を説明する。なお、ここでは、図11に示したユーザプログラムを実行するものとし、ロボット4が手先位置P0(図3参照)からコンベヤ1上を流れるワーク(動作指令が発行された時点でのワーク位置をP1:図3参照)の追従を開始してハンドリングし、ロボット4脇のテーブル上のパレット(位置P2:図3参照)に載置する場合を考える。ユーザは予め追従領域を追従領域記憶部15に設定しておき、その後、カメラ2でのワークの認識を行うプログラムを実行する。これによりカメラ2は撮像を開始し、その画像からワークの位置が求められてワーク現在位置記憶部13に記憶され、ワーク現在位置更新部14によってワーク現在位置記憶部13に逐次更新されている。この更新処理にあたっては、コンベヤ座標系での位置の演算が行われており、上述したように演算量は従来に比

べて格段に少なくなっている。また、イベント検出部17は、ワークの現在位置を追従領域記憶部15の追従領域設定値とをサンプリングタイム毎にチェックし、ワークが追従領域に到達してから追従領域から逸脱するまでの間、ワーク到達イベントを繰り返し発行し、逸脱すると、今度は追従停止イベントを発行する。以上の処理は、ユーザによりロボット動作プログラムが開始されていない間にもワークの認識に係るプログラムの実行によって開始されるもので、ロボット動作とは無関係にロボットコントローラ5側で行われている処理である。

【0089】ここで、ユーザが例えば図11に示したユーザプログラム(ロボット動作プログラム)を実行すると、ユーザプログラム実行部12は、タスク1及びタスク2を起動させる。そして、タスク1においては1行目においてワークの到達を待ち、そして、イベント検出部17からワーク到達イベントが発行されると、ユーザプログラム実行部12は次の命令に移行する。2行目は動作命令で、且つその位置データはコンベヤ1上の現在のワーク位置を指定する変数workposであるので、ユーザプログラム実行部12はフラグを1に設定し、上述した関節角度移動量 ΔJ 、時間Tを算出して軌道生成部16に動作命令の登録を行い、軌道生成部16からの動作命令終了通知待ち状態となる。

【0090】軌道生成部16は、登録されている動作命令に従う軌道の生成を行うが、ここでは指令位置がコンベヤ座標系で与えられているので、関節角度 $J_0(t)$ 及び追従動作分の関節角度移動量 $\Delta J_0(t)$ を計算しこれらを加算した目標軌道を生成し、その関節角度 $J(t)$ をモータパルス座標に変換して出力する。モータ制御部18は、軌道生成部16から出力されたパルスに従ってロボット4の各関節を制御する。軌道生成部16はこの軌道の生成を各サンプリングタイム毎に繰り返して行い、ハンドリングが終了すると、ユーザプログラム実行部に動作命令終了通知を発行する。

【0091】ユーザプログラム実行部12は、軌道生成部16からの動作命令終了通知を受信すると、次の命令に移行する。3行目においても動作命令であり、ここでは、ロボット座標系での位置P2であるため、ユーザプログラム実行部12は、フラグを0に設定すると共に関節角度移動量 ΔJ 及び時間Tを演算して動作命令の登録を行う。軌道生成部16は、ここで登録された動作命令に従って関節角度 $J_0(t)$ を算出し、ここでは追従動作分の関節角度移動量の計算は省略して前記関節角度 $J_0(t)$ をそのまま目標関節角度 $J(t)$ とし、モータパルス座標に変換して出力する。これにより、ロボット4はワークをハンドリング後も継続していた追従動作を停止しての位置P2への移動を行う。

【0092】一方、タスク2においては、1行目において追従停止イベント待ち状態にあり、イベント検出部17から追従停止イベントが発行された場合、2行目にお

いてタスク1を停止させ、3行目において追従動作を停止させるようになっている。

【0093】以上に説明したように、本実施の形態1によれば、コンベヤ座標系を定義することにより、ワークの現在位置の演算に係る演算量を低減でき、ロボットコントローラ5の負荷を軽減することができる。

【0094】また、コンベヤ座標系の位置データ（例えばコンベヤ座標系の位置データを示す変数又は座標系修飾「/CNV」を付与した位置データ）への動作命令を実行するだけで自動的に追従動作を開始し、逆にロボット座標系の位置データへの動作命令が実行されたときは、自動的に追従動作を停止するようにしているので、ユーザはコンベヤ1の動作を意識することなく、止まっているワークに対する処理を作成するように容易にプログラムすることが可能である。

【0095】また、コンベヤ座標系に対して追従領域を設定し、その追従領域へのワークの到達や追従領域からの逸脱をロボットコントローラ5側で（イベント検出部17で）検出してイベントとして発行するようにしたので、ユーザプログラムで検出するようにした従来の方法に比べてロボットコントローラ5の負荷を軽くでき、処理速度の低下が防止されて処理性能を向上させることが可能となる。また、ユーザはユーザプログラムにおいて処理ループを記述する必要がなくなり、イベント検出部17からのイベントの発行を待つ命令を記述するだけで良いため、プログラム記述が非常に楽になる。

【0096】また、そのプログラムは、図11に示したユーザプログラムの具体例に示されているように、各イベント毎にそのイベントが発生したときに実行させる処理をモジュール化することが可能となり、図20に示した従来のユーザプログラムと比較して明らかなようにプログラムの可読性が非常に向上している。

【0097】また、イベントの検出はソフトウェアで実現しているので、追従領域の変更等に対してもその追従領域設定値を変更するだけでよく、柔軟に対応することが可能となる。

【0098】また、ワークが追従領域内にないと判断した場合には追従停止イベントが発行されるので、ユーザは追従停止イベントに対応する処理として追従方向の軌道の生成を停止させるようにすれば、ロボットが追従領域を逸脱して追従動作を続ける等の不具合を確実に防止することが可能となる。また、ユーザはロボットが追従動作を行ったまま自分の動作範囲外にまで動作しようとしてエラーとなる等の不具合の回避動作プログラムを簡単に作成することが可能となる。また、ユーザプログラムによらず追従停止イベントが発行されたとき追従方向の軌道の生成を自動的に停止することも可能で、この場合、自動的に且つ確実に上記のような不具合を防止することが可能となる。

【0099】また、コンベヤ座標系は、コンベヤ1の移

動方向に平行なX軸、コンベヤ1の幅方向にY軸を有するように設定したので、コンベヤ1上のティーチングで、上流側に数mm、コンベヤ1の幅方向に数mmといったようなティーチングが容易に行えるようになる。

【0100】また、軌道の生成において、指令位置へのロボット4の軌道にコンベヤ1の動作量に係る追従方向の軌道を加算して目標軌道を生成しているので、コンベヤ1の速度ムラを補償した軌道が生成され、ワークを正確にハンドリングすることが可能となる。

【0101】実施の形態2. 図13は本発明の実施の形態2のハンドリングシステムを示す概略図である。本実施の形態2は、図3に示したカメラ2に代えてワーク（パレット23）の到着を検出する検出部であるセンサ21を備えたシステムである。このように構成したハンドリングシステムは、例えばコンベヤ1上のワーク（パレット23）がコンベヤ1の幅方向について整列されているような場合に構成されるシステムで、このシステムにおいて、例えばテーブル上のワーク22を縦横3つずつに等間隔に内部を仕切られたパレット23に移載する作業を行う場合、ユーザプログラムにおいてロボット手先の移動先としてパレット23内の各部屋を指定することになるが、この場合、コンベヤ座標系の導入により例えば図14に示すようにパレット11の左上端の部屋のコンベヤ座標系における位置A（ $x_A, y_A, 0$ ）をワーク現在位置として登録しておくことにより、例えばB点を指定する場合にはB（ $x_A + \Delta x, y_A - \Delta y, 0$ ）のように登録された現在位置からのオフセット（相対位置）によってその位置を指定することが可能となる。このようにコンベヤ座標系の導入によってコンベヤ1上を移動するワークに対する作業を簡単に記述することが可能となる。また、図13に示したハンドリングシステムは他に例えばパレット上に2種の部品が交互に並んでいて、それをコンベヤ1上で一体に組み立てるといった作業にも適用可能であり、この場合も上記実施の形態1とほぼ同様の作用及び効果が得られる。

【0102】実施の形態3. 図15は、本発明の実施の形態3のハンドリングシステムの概略図、図16は本実施の形態におけるコンベヤ座標系及び追従領域の説明図である。本実施の形態3は、図1におけるコンベヤ1をターンテーブル（円弧コンベヤも含む）31に替えたシステムであり、このようなターンテーブル31を利用する場合、コンベヤ座標系は図16に示すようにワークの回転方向（ターンテーブルの円周方向）にX軸、回転中心を通る半径方向にY軸を設定し、各座標値を（回転角度（°）、半径（mm）、高さ（mm））で指定する。また、追従領域を図のハッチングで示す領域に設定する場合、追従領域設定値はa（ $x_a, y_a, 0$ ）、b（ $x_b, y_b, 0$ ）となる。

【0103】このようなターンテーブル31の場合においても、ワークの現在位置はx座標値のみを更新すれば

良く、実施の形態 1 及び実施の形態 2 とほぼ同様の作用及び効果を得ることが可能となる。

【0104】実施の形態 4. 図 17 は本発明の実施の形態 4 を示すハンドリングシステムの概略図である。このハンドリングシステムは、1 台のコンベヤに対して複数のロボットを設置するもので、この場合、ロボット毎にコンベヤ座標系及び追従領域を設定する。このシステムは例えば複数のワークが高速で流れており、ロボット 4 で取り損ねたワークをロボット 43 でハンドリングしたり、異なる 2 種のワークが混ざって流れている場合に、一方のワークをロボット 4 で、他方のワークをロボット 43 でハンドリングしたりといった作業を行うものであり、このようなハンドリングシステムにおいても、本発明を適用することにより、上記実施の形態 1 及び実施の形態 2 とほぼ同じ作用及び効果を得ることが可能となる。

【0105】実施の形態 5. 図 18 は本発明の実施の形態 5 を示すハンドリングシステムの概略図である。このシステムは、複数のコンベヤに対して 1 台のロボットを設置するもので、この場合、コンベヤ毎にコンベヤ座標系を定義することにより、例えばコンベヤ 1 からコンベヤ 1 a にワークを移載する作業等も簡単にプログラム可能となる。

【0106】なお、コンベヤ座標系には、上記したもの以外に、例えば図 19 に示すような①斜行座標系、②それ以外の座標系も考えられる。これらの場合の追従領域は、図 19 のハッチングで示された領域であり、追従領域設定値は図 6 に示した直交座標系と同様に、対角上の 2 点の座標値 $a(xa, ya, 0)$ 、 $b(xb, yb, 0)$ となる。

【図面の簡単な説明】

【図 1】本発明の実施の形態 1 のロボットコントローラを適用したシステムの機能ブロック図である。

【図 2】本発明の実施の形態 1 のソフトウェア構成を示す図である。

【図 3】本発明の実施の形態 1 のシステムの概略構成図である。

【図 4】コンベヤ座標系の説明図である。

【図 5】各座標系の階層構造を示す図である。

【図 6】追従領域の説明図である。

【図 7】図 1 のイベント検出部の動作フローを示す図である。

ある。

【図 8】図 1 のユーザプログラム実行部の動作フローを示す図である。

【図 9】本発明の実施の形態 1 における位置関係の説明図である。

【図 10】図 1 の軌道生成部の動作フローを示す図である。

【図 11】本発明の実施の形態のユーザプログラムの一例を示す図である。

【図 12】図 11 のタスク 1 及びタスク 2 の処理フローを示す図である。

【図 13】本発明の実施の形態 2 のハンドリングシステムの概略図である。

【図 14】図 13 の実施の形態 2 おけるワークの位置関係の説明図である。

【図 15】本発明の実施の形態 3 のハンドリングシステムの概略図である。

【図 16】図 15 の実施の形態 3 におけるコンベヤ座標系及び追従領域の説明図である。

【図 17】本発明の実施の形態 4 のハンドリングシステムの概略図である。

【図 18】本発明の実施の形態 5 のハンドリングシステムの概略図である。

【図 19】コンベヤ座標系及び追従領域の他の例を示す図である。

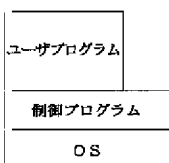
【図 20】従来のユーザプログラムの具体的一例を示す図である。

【図 21】図 20 の処理フローを示す図である。

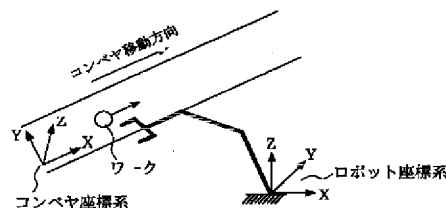
【符号の説明】

- 1 コンベヤ
- 2 カメラ
- 4 ロボット
- 5 ロボットコントローラ
- 12 ユーザプログラム実行部
- 13 ワーク現在位置記憶部
- 14 ワーク現在位置更新部
- 15 追従領域記憶部
- 16 軌道生成部
- 17 イベント検出部
- 21 センサ
- 31 ターンテーブル

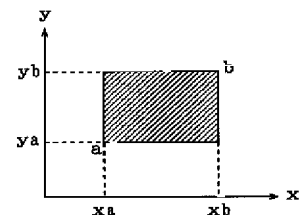
【図 2】



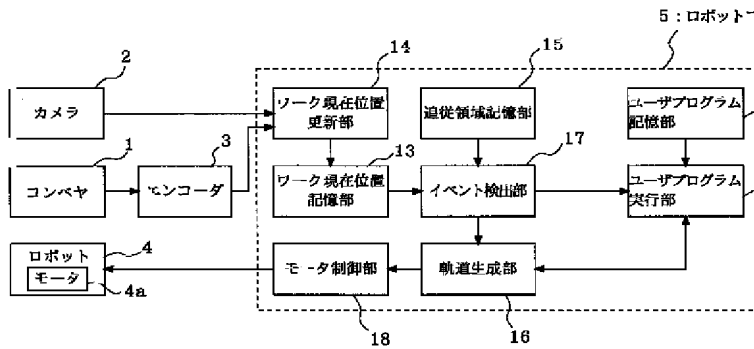
【図 4】



【図 6】

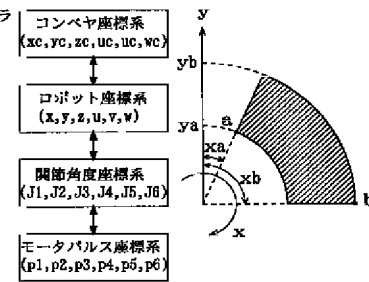


【図1】

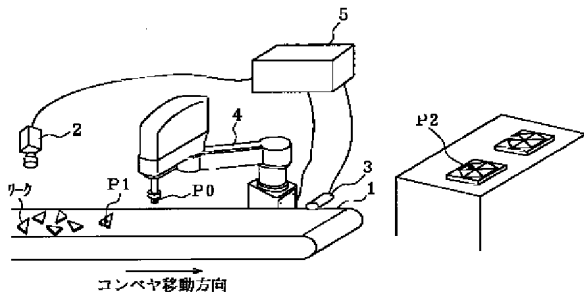


【図5】

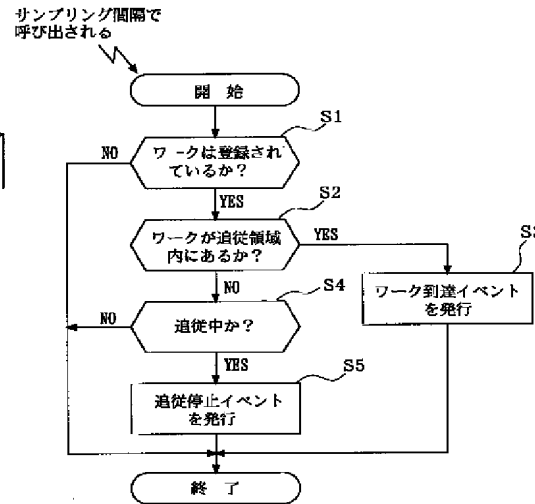
【図16】



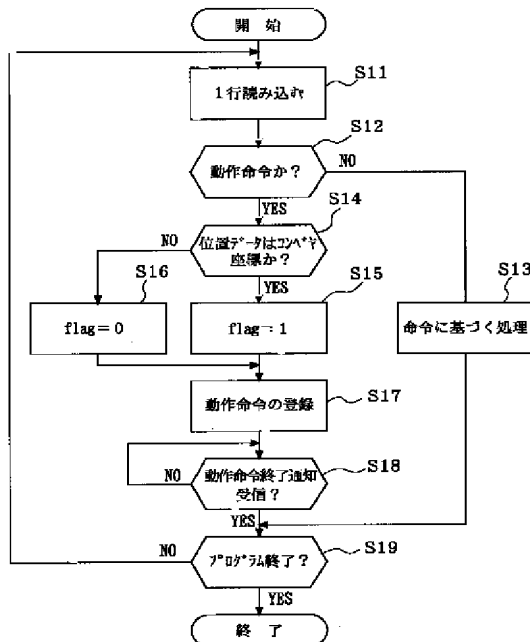
【図3】



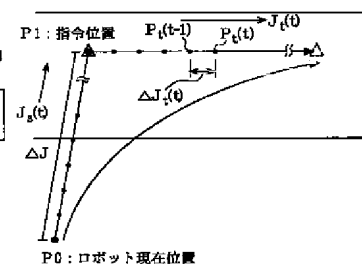
【図7】



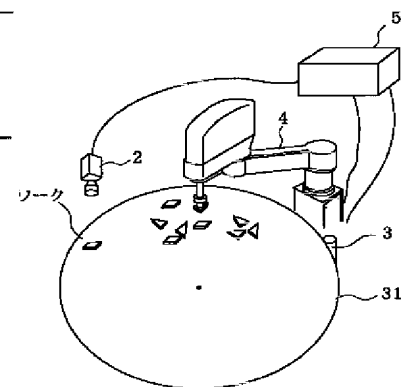
【図8】



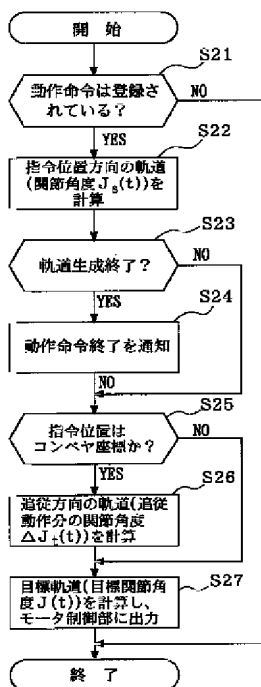
【図9】



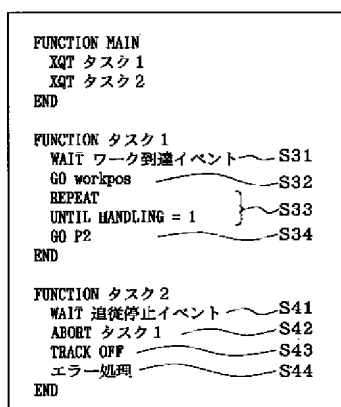
【図15】



【図10】

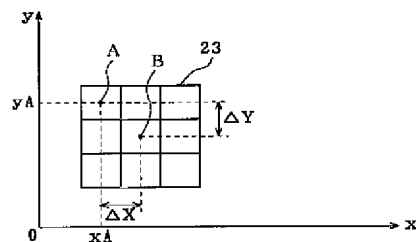


【図11】

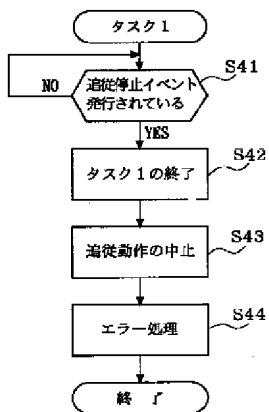
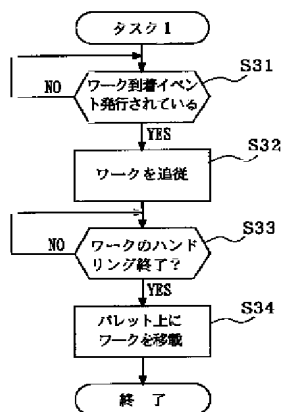


ここで、
 XQT タスク名：タスク名で指定されるプログラムを起動する命令
 ABORT タスク名：タスク名で指定されるプログラムを終了させる命令とする。

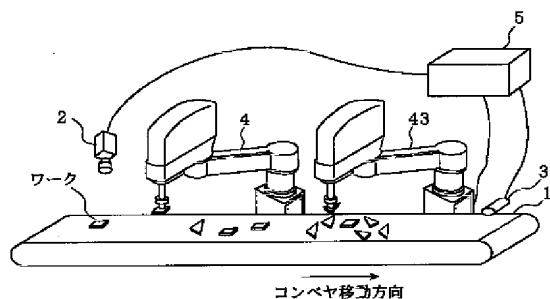
【図14】



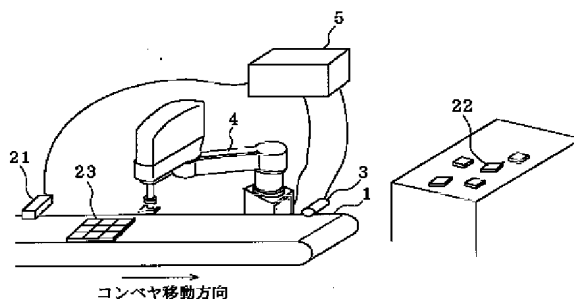
【図12】



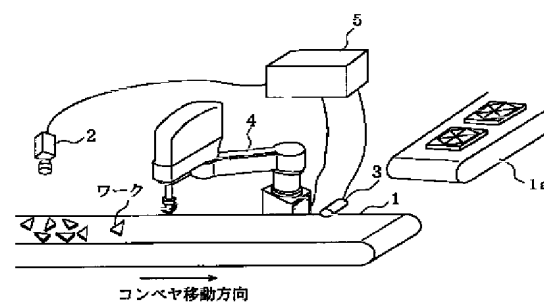
【図17】



【図13】

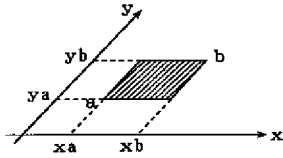


【図18】

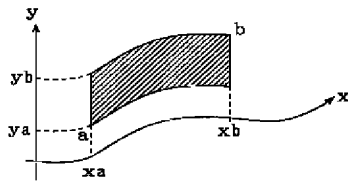


【 図 19 】

① 斜交座標系



② それ以外



【 図 20 】

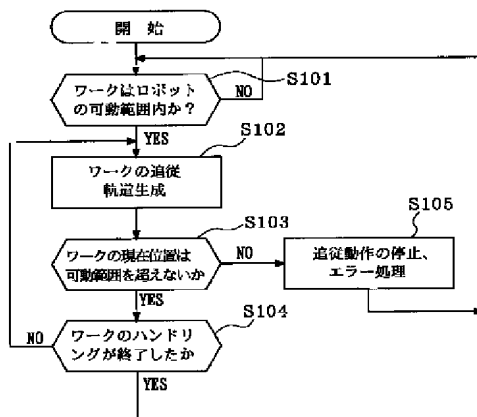
```

FUNCTION MAIN
DO
  IF ( workpos.x >= 100 ) AND ( workpos.x <= 200 )
    AND ( workpos.y >= 300 ) AND ( workpos.y <= 400 ) THEN } S101
    JUMP workpos ~ S102
  eflag = 0
  hflag = 0
  REPEAT
    TRACK ON
    IF ( nowpos.x >= 100 ) AND ( nowpos.x <= 200 )
      AND ( nowpos.y >= 300 ) AND ( nowpos.y <= 400 ) THEN } S103
      eflag = 0
    ELSE
      eflag = 1; hflag = 1
    ENDIF
    IF HANDLING = 1 THEN hflag = 1 ~ S104
  UNTIL hflag = 1
  TRACK OFF
  IF eflag = 1 THEN エラ - 処理 ~ S105
  LOOP
END

```

ここで、workpos.x: ワークの現在位置の X 座標値
 workpos.y: ワークの現在位置の Y 座標値
 追従作業を行う領域: (100,300)-(200,400)
 TRACK ON/OFF: コンベヤへの追従を行う/行わない命令
 HANDLING: ハンドリング終了したかどうかを返す関数。終了で 1 を返す

【 図 21 】



フロントページの続き

(72) 発明者 亀山 孝之
 千葉県習志野市屋敷 4 丁目 3 番 1 号 セイ
 コー精機株式会社内
 (72) 発明者 説田 信之
 千葉県習志野市屋敷 4 丁目 3 番 1 号 セイ
 コー精機株式会社内

F ターム (参考) 3F059 AA05 AA13 BA04 BB02 DB04
 FA03 FB12 FC01 FC13
 3F060 AA01 AA03 CA24 EB02 EB12
 EC02 EC12 GD14
 3F072 GA03 GA10 GD01 GE01 GE08
 GG01 KB03 KD07 KD23 KD28
 5B057 AA02 BA02 DA07